Refine Search

Search Results -

Terms	Documents
L47 and (domain with object or domain near object or domain adj object)	3

Database: US Pre-Grant Publication Full-Text Database; US Patents Full-Text Database; US OCR Full-Text Database; EPO Abstracts Database; JPO Abstracts Database; Derwent World Patents Index; IBM Technical Disclosure Bulletins

Search:

Refine Search

Recall Text | Clear | Interrupt

Search History

DATE: Wednesday, November 30, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L48	L47 and (domain with object or domain near object or domain adj object)	3	L48
L47	(phonetic near value or phonetic with value or phonetic adj value)	412	L47
L46	L45 and (phonetic near value or phonetic with value or phonetic adj value)	2	L46
L45	L44 and nodes	299	L45
L44	L43 and (hierarchy or hierarchical or hierarch$)	520	L44
L43	L42 and (character with length or character near length or character adj length)	1230	L43
L42	L41 and value	14764	L42
L41	(text near object or text with object or text adj object)	22553	L41
L40	L39 and (grammar with rules or grammar near rules or grammar adj rules)	6	L40
L39	l37 and semantic same information	16	L39
L38	L37 and (text with object or text near object or text adj object)	5	L38
L37	syntactic near2 (hierarch$ or hierarchy or hierarchical)	37	L37
L36	704/8	703	L36
L35	L34 and syntactic near2 (hierarch$ or hierarchy or hierarchical)	2	L35
L34	text near object	4388	L34

L33	L32 and (index or indices or indises)	444	L33
L32	directory near service and semantic	773	L32
L31	717.clas.	9812	L31
L30	717/107	309	L30
L29	704/9	1515	L29
L28	704.clas.	19953	L28
L27	715/541	70	L27
L26	715/531	313	L26
L25	715/513	1718	L25
L24	715.clas.	23353	L24
L23	707.clas.	30753	L23
L22	707/541	152	L22
L21	707/531	908	L21
L20	707/513	2484	L20
L19	707/102	6642	L19
L18	707/101	4225	L18
L17	707/100	6945	L17
L16	707/9	2571	L16
L15	707/4	4416	L15
L14	707/3	7673	L14
L13	707/1	7228	L13
L12	707/1	7228	L12
DB=USPT; PLUR=YES; OP=OR			
L11	(5434974 | 5428777 | 5227971 | 5333317 | 5265065 | 4974191 | 5276616 | 5268839)! [PN]	8	L11
DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L10	('5515534')[ABPN1,NRPN,PN,TBAN,WKU]	2	L10
L9	5515534.pn.	2	L9
DB=USPT; PLUR=YES; OP=OR			
L8	(4631673 | 5649186 | 5748954 | 5737592 | 5706507 | 4918593 | 4918588 | 5761663 | 5307484 | 5386556)![PN]	10	L8
DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L7	('5826258')[ABPN1,NRPN,PN,TBAN,WKU]	2	L7
L6	5826258.pn.	2	L6
DB=USPT; PLUR=YES; OP=OR			
L5	(5111398 | 5424947 | 5406480 | 5146406 | 5386556)![PN]	5	L5
DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L4	('5966686')[ABPN1,NRPN,PN,TBAN,WKU]	2	L4
DB=USPT; PLUR=YES; OP=OR			
L3	'5406480'.pn.	1	L3
L2	'5406480'.pn.	1	L2
DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L1	5966686.pn.	2	L1

END OF SEARCH HISTORY

L40: Entry 5 of 6 File: USPT Sep 28, 1999

DOCUMENT-IDENTIFIER: US 5960384 A

**** See image for Certificate of Correction ****

TITLE: Method and device for parsing natural language sentences and other sequential symbolic expressions

Brief Summary Text (5):

Automated sentence parsing devices--usually general purpose digital computers--have operated by matching parts of the input sentence with a very large number of stored rules (Tennant et al., 1989, U.S. Pat. No. 4,829,423). If the first rule does not apply, the next is chosen for examination. Three broad classes of devices have been used; these employ increasingly deep levels of sentence analysis. The most common approach is Syntactic Parsing, in which the rules driving the parsing device describe the organization of words in a particular language. In practice, achieving a parse also requires resolving the possible senses of ambiguous words. This is done by a semantic interpreter, which uses information about word meaning. Semantic Parsing aims at the level of meaning directly, driving the parser with rules relating ordered words to the underlying meaning of the sentence. Conceptual Analysis also aims directly for the level of meaning, but does not attempt to preserve word order. Conceptual analysis parsers identify key words as tokens for the underlying objects or actions, and the organization sought is that of the objects being discussed rather than that of the words used to discuss them. The processor reconstructs this organization by using "encyclopedia information" about the objects, rather than "dictionary information" about the words.

Brief Summary Text (10):

Hierarchical structures can be constructed generatively, however. Recursive Transition Networks allow transition networks to accomplish a hierarchical parse by allowing parser transitions to be triggered not only by lexical categories like "adjective," but also by syntactic categories such as "noun phrase" (Winograd, 1983, pp. 196-198). Deciding whether the input sentence contains a noun phrase, however, requires first activating additional transition networks for describing the constituents of noun phrases. Immediate-Constituent Grammar parsers build the tree-like structure by using a list of rules that relate two levels of structure (Winograd, 1983, pp. 75-76). The list includes rules linking sentences to noun-, verb-, and prepositional phrases; these to lexical categories such as noun, adjective, and determiner; and these to specific words. An example is: sentence.fwdarw.noun phrase+verb phrase; noun phrase.fwdarw.determiner+adjective+noun; verb phrase.fwdarw.verb+adverb+direct object. Successive application of the rules produces a tree of increasing detail: parsing a sentence is achieved by successively checking each rule in the list for fit to the input sequence of words, and choosing from the rule list those that match. One list of rules, one grammar, is capable of building many different trees. The particular tree that results depends on the subset of rules that matched the particular sentence's words.

Brief Summary Text (11):

In Role Structures parsers, the syntactic categories are themselves in effect constituents of a pattern of syntactic roles (such as the pattern "subject-object") (Winograd, 1983, pp. 79-80). Word tags can also be processed generatively (Hutchins, 1991, U.S. Pat. No. 4,994,966). A related strategy is to copy the dictionary entries of the words of a sentence directly into memory, and then use the grammar rules to directly remove properties from, or add to, these dictionary entries (Zamora et al., 1989, U.S. Pat. No. 4,887,212).

Brief Summary Text (12):

A disadvantage of the above devices is that they use context-free grammars and are thus inadequate for sentences that have long-distance dependencies. Such dependencies arise from number/case agreement, passive-voice constructions, and questions. Transformational Grammars use a series of context-sensitive rewrite rules to transform a context-free syntactic tree

structure into another tree that matches the word order found in, for example, the passive voice (Winograd, 1983, pp. 139-143). However, these processes have not been widely incorporated into devices (Winograd, 1983, p. 162).

Brief Summary Text (13):
Phrase Structure Grammar uses immediate-constituent rules, but adds "derived constituents" of the form "verb phrase but missing x." For example, a derived constituent like VP/NP carries the information that a noun phrase is missing from the verb phrase. Long-distance dependencies are resolved by using semantic knowledge to find an NP that fills in this hole (Winograd, 1983, pp. 345-346). The strategy most frequently used in practical parsers is the Augmented Transition Network (Loatman et al., 1990, U.S. Pat. No. 4,914,590). Here, the recursive transition network's set of allowed transitions between parser states is supplemented with registers that record a word's features, such as number or gender, and a word's role, such as subject, object, or determiner. Rules governing the allowed transitions are made dependent on the condition of these registers (Winograd, 1983, pp. 204-210). In this way, long-distance dependencies are incorporated. Charts are kept of the intermediate states of the parse to improve efficiency.

Brief Summary Text (14):
A completely different generative approach is Tree-Adjoining Grammar, which combines not rules but trees (Joshi, 1985). Each initial tree, actually a "mini-tree," describes a sentence fragment. These are combined to build a tree for the complete sentence. By collecting several rules into one mini-tree, a long-distance dependency can be incorporated from the outset as a mini-tree having links between two of its branches. However, since each mini-tree represents just one of the combinations of several rules, the initial stock of mini-trees is very large. The number of tree-choice decisions is accordingly large and the parsing time is proportional to the fourth power of sentence length. A "molecular parsing" device has been described in which each word's dictionary entry contains both phrase structure rules and template-patterns for the word's possible successors (Hu, 1994, U.S. Pat. No. 5,297,040).

Brief Summary Text (15):
Implementing these syntactic parsing approaches has not been straightforward, for two reasons. First, a correct parse requires a single grammar rule to be chosen at each word. But the correctness of a rule is usually not evident until several words later. Therefore, the large number of rules required to capture real sentences means that the device will pursue many fruitless parsing paths, followed by backtracking when the parse becomes clearly incorrect. This rule-choice problem has fostered strategies to minimize the time spent on fruitless parsing paths, including pursuing multiple parses in parallel or retaining information about partially-successful parses (Winograd, 1983, pp. 368-369).

Brief Summary Text (19):
More direct reduction of word ambiguity, and rule choice, has been sought by associating word-meaning information with each element of the syntactic parse tree after the syntactic tree is constructed. Semantic interpretation then usually uses a Pattern-Matching strategy against stored semantic template-patterns. These stored patterns can be of several types. Selection Restriction Rules specify the objects with which a given word may have a relation; these are embodied as arguments to the word's entry in the dictionary, or "lexicon." An example is: "green"+(physical object).fwdarw.green color (Allen, 1995, pp. 295-296). Rather than denoting a meaning, these rules lead to a choice among previously-cataloged meanings. Semantic Network patterns are hierarchies of object-types or action-types, based on semantic primitives, which the designer constructs in advance. If the semantic network also includes "semantic functions," such as "agent," "instrument," or "result," it can be used to limit the possible senses of a word (Allen, 1995, pp. 305-307). In actual devices, such hierarchies are implemented as a nested list.

Brief Summary Text (21):
Other strategies are less often used in practical parsing devices. Systemic Grammar uses hierarchies of template-patterns based not only on semantic functions, but also on word and pragmatic functions. The functions are arranged into systems organized as a tree of choices: mood (question, statement, command); transitivity (actor, action, goal; agent, instrument, result); theme (focus of attention); and information (already-given, new) (Allen, 1995, p. 95; Beardon, 1992, pp. 187-188). These patterns are reflected in the sentence's word order and lexical-category patterns. Thus, a degree of semantic information emerges once the correct lexical category pattern is matched.

Brief Summary Text (23):
Several parsing devices use Generative methods, building semantic structures that correspond to the syntactic parse tree. The semantic rules are incorporated within the syntactic rules that the syntactic parser uses. Of these, Definite Clause Grammar supplements the immediate-constituent grammar rules with further information about word features, possible word roles in the sentence, and semantic functions (Winograd, 1983, p. 349). These rules are expressed in predicate calculus; parsing then uses a theorem-prover algorithm.

Brief Summary Text (24):
Lexical-Functional Grammar supplements a syntactic tree with a semantics-like tree (Winograd, 1983, p. 334-337; Tokuume et al., 1992, U.S. Pat. No. 5,101,349). To achieve this, the designer first identifies rules governing the inheritance of word features, word roles, and semantic functions down a context-free constituent-structure tree. He assigns to each constituent in an immediate-constituent rule an equation that describes the relation between the features, roles, and functions of the parent node and those of the child node. Thus, the parser will build a feature/role/function tree as it builds the syntactic tree. In addition, the parser uses a detailed dictionary that includes feature/role information about each word and its possible arguments. This word information will then percolate up through the feature/role/function tree. The rules for the syntactic tree can be simple; though they would allow many incorrect sentences, the constraint resulting from solving the set of feature/role/function equations results in a correct parse. The parser solves this set of simultaneous equations instead of pursuing all of the parsing paths made possible by an ambiguous word.

Brief Summary Text (27):
Montague Semantics asserts that both the syntax and semantics of natural languages obey formal laws (Dowty, 1981, pp. 180-183). Each compositional syntactic rule is associated with a parallel compositional semantic rule. The syntactic analysis uses a phrase-structure grammar's derived categories (see above) to construct a syntactic tree, but with a very large hierarchy of syntactic constituent-types that corresponds to the hierarchy of logical types found in the predicate calculus of the semantics. In the semantics portion, the meaning of a word is defined as being the word's referent--specified by membership in sets of individuals or sets of properties. Montague semantics focusses on building precise specifications of set-membership. In principle, the syntactic analysis would automatically generate the semantic representation. But focussing on set membership does not address word ambiguity, and the approach appears to have been used only on small subsets of English.

Brief Summary Text (31):
A Generative approach to semantic parsing is Semantic Grammar, in which immediate-constituent rules use semantic-like categories or special words (Allen, 1995, pp. 332-334; Beardon et al., 1992, pp. 129-130). A set of rules for airline reservations might be: reservation-phrase.fwdarw.reservation+reservation-modifiers; reservation.fwdarw.reservation-verb+flight#; reservation-modifiers.fwdarw."for"+name. Such rules are domain-specific. Also, the parser must have a separate rule for each syntactic variant, such as passive voice.

Brief Summary Text (33):
More ambitious is the attempt to parse a sentence according to the concepts it represents. This is done by viewing words as tokens for the underlying objects or concepts. The processor assembles this information by using "encyclopedia information" about the objects, rather than "dictionary information" about the words. Because words are of course the input for such parsing, even though it is their referents that are being analyzed, conceptual analysis can be difficult to discern from semantic parsing. A useful diagnostic is that conceptual analysis does not attempt to preserve word order. Consequently, conceptual analysis devices are not true natural language parsers.

Brief Summary Text (38):
First, syntactic procedures identify the main verb, the main noun, and the words and phrases that depend on these syntactically (Wilks, 1976, pp. 169-173). However, a syntactic tree is not produced. Instead, the lexicon is used to replace the verb with a conceptual dependency structure based on conceptual primitives; this structure is annotated with conceptual case information, similar to semantic case. The conceptual cases predict other conceptual items expected in the sentence, such as agents and recipients, for which the parser then searches. These are found by comparing unused sentence elements to syntactic patterns that have been prestored.

Brief Summary Text (40):
The disadvantages of present parsing methods are thus complexity, large required processor size, limited vocabulary, and slow speed. These disadvantages stem from the large number of syntactic rules used and the incorporation of semantic information to resolve ambiguities.

Brief Summary Text (41):
What is desired, therefore, is a parsing method and device which would utilize a small number of rules, minimal semantic information, and minimal computing power. Such parsing method and device would require minimal disambiguation of words having multiple meanings and would not rely on specific knowledge bases, word co-occurrence probability data, selection-restrictions, frames, or expectations about sentence content. Such parsing method and device would also process symbol strings in a time proportional to sentence length.

Brief Summary Text (44):
Another object of the invention is to provide a system to parse thought expressions, such as natural language, using minimal semantic information.

Brief Summary Text (56):
When we see a squirrel burying nuts, we are aware of the squirrel, the nuts, and the activity of burying. Similarly, the sentence "Squirrels bury nuts" can be divided into picture-relation-picture form: (Squirrels) bury (nuts). The verbal relation "bury" acts as a parsing element in languages that use this word order. The picture-relation-picture form, hereafter abbreviated "P rel P," can itself be a "picture", (P rel P). Such a picture is referred to herein as a "picture-symbol group"; e.g., ((Squirrels) bury (nuts)). More complex sentences can be shown to have the same (P rel P) form, with various other types of relations such as prepositions acting as parsing elements: ((Squirrels) bury (nuts)) under (trees). [((Squirrels) bury (nuts)) under (trees)] when [(it) is (Fall)]. The square parentheses are used for clarity and do not differ in principle from ordinary parentheses. These structures can be written using an immediate-constituent rule: P rel P.fwdarw.P. This rule differs from those of immediate-constituent grammars in the prior art because i) it generates a tri-forked tree, rather than the bifurcated tree which the standard subject-predicate division generates and ii) each level of the tree uses the same immediate-constituent rule.

Brief Summary Text (87):
The parser uses the (P rel P) well-formedness criterion to regenerate missing relations in several situations. When two successive picture-words appear in a sentence, we know only that the two P's were associated in the speaker's mind. The (P rel P) criterion would tell us where to insert the minimal relation "associated with." However, the relation omitted in English seems to be always either.right brkt-top. or.left brkt-top.. Knowing which P is more specific determines the choice between the two. Thus missing relations can be regenerated by employing primitive semantic information.

Brief Summary Text (161):
Cognitive parsers are additionally distinct from syntactic parsers supplemented by semantic interpretation because they do not use pre-constructed semantic networks, set membership hierarchies, or word-based selection-restriction rules. Indeed, they are insensitive to most semantic information: they do not distinguish between "The squirrel buried a nut" and "My aunts sent a telegram," and they allow "Colorless green ideas sleep furiously." For the same reason, cognitive parsers are distinct from devices that construct predicate-argument structures. Cognitive parsers are further distinguished from many prior-art semantic interpretation parsers in being generative rather than matching template-patterns for semantic case, mood, transitivity, theme, information, or voice. Cognitive parsers do not build parsed structures by assembling overlapping parse fragments.

Brief Summary Text (162):
Cognitive parsers are distinct from semantic parsers because they are insensitive to most semantic information; as noted above, they do not distinguish between "The squirrel buried a nut" and "My aunts sent a telegram." Cognitive parsers are additionally distinct from semantic parsers because they are not limited to parsing words; sequences of icons or sign language configurations can also be parsed.

Brief Summary Text (163):
Cognitive parsers are distinct from conceptual analyzers because of their insensitivity to most semantic information. They are additionally distinct from conceptual analyzers because they

preserve word order. Cognitive parsers furthermore do not use pre-constructed templates for commonly-encountered messages; conceptual selection-restriction rules; or conceptual case structures. They also do not seek key words on which to base slot-filler structures.

Detailed Description Text (53):
FIG. 13B shows the structure of a natural language translation device, for which the input symbol sequence 2 is a natural language text. The cognitive parser 170 produces a parsed symbol sequence 8, using a syntactic and semantic analyzer unit 176 to resolve words that can be either a picture or a relation, rather than using syntactic and semantic analysis to decide among grammar rules. The parsed symbol sequence 8 is then the input into a natural language translation unit 178, which translates the expression in the first language to an equivalent expression in a second language and displays the result.

Detailed Description Text (57):
The cognitive parser provides a simple means of parsing natural language and other symbolic expressions of thought using a small number of stored rules, minimal stored semantic information, and minimal requirement for disambiguation, and is capable of parsing while the symbolic expression is being entered into the device. Its operation is not limited to a specific knowledge base or to anticipatable thoughts. The cognitive parser's simplicity allows it to be used in personal computers and in non-computer devices equipped with only minimal computing power.

Detailed Description Text (58):
The parsed natural language structure produced would make it easier for untrained users to use a computer for storing information, querying a database, or finding logical inferences. By replacing multiple layers of menus, it would make computers easier to use even for trained users and allow control of kitchen appliances, videocassette recorders, home lighting, and heating and air conditioning. Another use for the parsing operations would be as a programming language that resembled natural language. Because cognitive form reflects the structure of the thoughts underlying language more accurately than does traditional grammar, the cognitive parser would be useful in teaching grammar. Because those sentences which can be parsed using cognitive rules without semantic analysis are easiest to understand, the cognitive parser could be used as a reading-level checker. In combination with a device performing semantic interpretation, lexical-functional grammar, or preference semantics, the cognitive parser could parse non-grammatical or idiomatic sentences. Combination with a device performing semantic interpretation or conceptual dependency analysis could constitute an input to a device performing logical inference involving the properties of objects.

Previous Doc Next Doc Go to Doc#